Attorney Docket No. 826.1921

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Yoshiki Okumura

Application No.: NEW

Group Art Unit: Not Yet Assigned

Filed: February 20, 2004

Examiner: Not Yet Assigned

For: ARITHMETIC DEVICE FOR MULTIPLE PRECISION ARITHMETIC FOR MONTGOMERY MULTIPLICATION RESIDUE ARITHMETIC

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN APPLICATION IN ACCORDANCE WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No(s). 2003-46527

Filed: February 24, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: February 20, 2004

By: _____
J. Randall Beckers
Registration No. 30,358

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

JAPAN PATENT OFFICE

This is to certify that the annexed is a true copy of the following application as filed with this office.

Date of Application: February 24, 2003

Application Number: Patent Application No. 2003-046527
[ST.10/C] [JP2003-046527]

Applicant(s): FUJITSU LIMITED

October 14, 2003

Commissioner,
Japan Patent Office Yasuo IMAI

Certificate No.P2003-3084342

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 Date of Application:	２００３年　２月２４日
出願番号 Application Number:	特願２００３－０４６５２７
[ST. 10/C]:	[ＪＰ２００３－０４６５２７]
出願人 Applicant(s):	富士通株式会社

２００３年１０月１４日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫

出証番号　出証特２００３－３０８４３４２

【書類名】　特許願

【整理番号】　0253273

【提出日】　平成15年 2月24日

【あて先】　特許庁長官殿

【国際特許分類】　G09C　1/00

G06F　7/72

H04L　9/00

【発明の名称】　モンゴメリ乗算剰余の多倍長演算のための演算装置

【請求項の数】　5

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　株式会社富士通コンピュータテクノロジ内

【氏名】　奥村　嘉樹

【特許出願人】

【識別番号】　000005223

【氏名又は名称】　富士通株式会社

【代理人】

【識別番号】　100074099

【住所又は居所】　東京都千代田区二番町８番地２０　二番町ビル３Ｆ

【弁理士】

【氏名又は名称】　大菅　義之

【電話番号】　03-3238-0031

【選任した代理人】

【識別番号】　100067987

【住所又は居所】　神奈川県横浜市鶴見区北寺尾７－２５－２８－５０３

【弁理士】

【氏名又は名称】　久木元　彰

【電話番号】　045-573-3683

【手数料の表示】

【予納台帳番号】　012542

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9705047

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　モンゴメリ乗算剰余の多倍長演算のための演算装置

【特許請求の範囲】

【請求項１】　ビットパターンで表された被乗数Ａと乗数Ｂの乗算を行う演算装置であって、

被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2i}、およびb_{2i-1}を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、ｉの数と同じ数だけの部分積を加算して、乗算結果を生成する加算手段とを備え、

前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの２の補数を生成して、該被乗数Ａの２の補数を前記乗算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【請求項２】　ビットパターンで表された被乗数Ａと乗数Ｂの乗算を行う演算装置であって、

被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2i}、およびb_{2i-1}を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、ｉの数と同じ数だけの部分積を加算して、乗算

結果を生成する加算手段とを備え、

前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの１の補数を生成して、該被乗数Ａの１の補数を前記乗算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【請求項３】　ビットパターンで表された被乗数Ａと乗数Ｂを乗算し、乗算結果とビットパターンで表された数Ｃおよび数Ｄとを加算する、乗算加算を行う演算装置であって、

被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2i}、およびb_{2i-1}を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、ｉの数と同じ数だけの部分積と数Ｃと数Ｄとを加算して、乗算加算結果を生成する加算手段とを備え、

前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの１の補数を生成し、該被乗数Ａの１の補数と数Ｃと数Ｄとを加算した結果を前記乗算加算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【請求項４】　ｋビットのビットパターンで表された被乗数Ａと乗数Ｂを乗算し、乗算結果とｋビットのビットパターンで表された数Ｃおよび数Ｄとを加算することで、１ブロックがｋビットである$g+1$個のブロックからなる整数Ｙから１ブロックがｋビットであるg個のブロックからなる整数Ｎを減算する演算を行う演算装置であって、

被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部

分積生成手段と、

乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2i}、およびb_{2i-1}を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、ｉの数と同じ数だけの部分積と数Ｃと数Ｄとを加算して、２ｋビットの乗算加算結果を生成する加算手段と、

前記乗算加算結果の一部のビットを反転する反転手段とを備え、

前記部分積生成手段が、整数Ｎのｊ番目のブロックn_jを被乗数Ａとして用い、前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの１の補数を生成し、ｊ－１番目のブロックの乗算加算からのキャリーを数Ｃとして用い、整数Ｙのｊ番目のブロックy_jを数Ｄとして用いて、該被乗数Ａの１の補数と数Ｃと数Ｄとを加算した結果をｊ番目のブロックの乗算加算結果として出力し、前記反転手段が、該ｊ番目のブロックの乗算加算結果の一部のビットを反転してｊ＋１番目のブロックの乗算加算へのキャリーを生成することを特徴とする演算装置。

【請求項５】　ビットパターンで表された整数ＩおよびＪと剰余の法Ｎを１ブロックがｋビットであるｇ個のブロックにそれぞれ分割して、$Y = IJ^{2^{-kg}} \bmod N$となるモンゴメリ乗算剰余の多倍長演算を行う演算装置であって、

ｋビットの被乗数Ａ、乗数Ｂ、数Ｃ、および数Ｄのそれぞれについて、与えられた複数の値のいずれかを選択して出力する第１の選択手段と、

前記第１の選択手段から出力される被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

前記第１の選択手段から出力される乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2}

ｉ、およびｂ２ｉ－１を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する第２の選択手段と、

前記第２の選択手段から出力される、ｉの数と同じ数だけの部分積と、前記第１の選択手段から出力される数Ｃと数Ｄとを加算して、２ｋビットの乗算加算結果を生成する加算手段と、

前記乗算加算結果の一部のビットを反転する反転手段とを備え、

前記第１の選択手段が、整数Ｎのｊ番目のブロックｎｊを被乗数Ａとして選択し、ｊ－１番目のブロックの乗算加算からのキャリーを数Ｃとして選択し、Ｙのｊ番目のブロックｙｊを数Ｄとして選択し、前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの１の補数を生成し、該被乗数Ａの１の補数と数Ｃと数Ｄとを加算した結果をｊ番目のブロックの乗算加算結果として出力し、前記反転手段が、該ｊ番目のブロックの乗算加算結果の一部のビットを反転してｊ＋１番目のブロックの乗算加算へのキャリーを生成するような動作モードを設けたことを特徴とする演算装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、モンゴメリ乗算剰余の多倍長演算を行うための演算装置に関する。

【０００２】

【従来の技術】

近年、コンピュータネットワーク上では、セキュリティ対策として、暗号化データ通信やユーザ認証（デジタル署名）が多く使用されている。これらのセキュリティ対策では、ＲＳＡ（Rivest-Shamir-Adleman）暗号や楕円曲線暗号等の暗号処理が利用される。

【０００３】

ＲＳＡ暗号や楕円曲線暗号は非常に大きいビット数の数値を扱うため、これらの暗号処理で使われるモンゴメリ乗算剰余の演算では、ソフトウェアやハードウェアで実現し易い多倍長演算アルゴリズムがよく使われる。多倍長演算アルゴリズムとは、非常に大きいビット数の数値を、ソフトウェアやハードウェアで扱い易いビット数（ブロック）に分割して、ブロック単位の演算を繰り返すアルゴリズムである。

【０００４】

モンゴメリ乗算剰余の多倍長演算アルゴリズムとしては、図１２のようなアルゴリズムが知られている（例えば、特許文献１参照）。図１２において、ＡおよびＢは整数であり、Ｎは剰余の法であり、Ｙは演算結果である。図１２の左側に示すアルゴリズムは、その右側に示すように変形することができる。このアルゴリズムは、$Y = AB2^{-kg} \mod N$の演算を表しており、①～⑥の処理からなる。

【０００５】

ｇはＡ、Ｂ、およびＮをｋビット＝１ブロックで分割した時のＡ、Ｂ、およびＮのブロック数であり、Ｙのブロック数はｇ＋１である。a_i、b_i、n_i、およびy_iはそれぞれＡ、Ｂ、Ｎ、およびＹのｉ番目のブロックを表す。Ａ、Ｂ、Ｎ、およびＹは、それぞれ、a_i、b_i、n_i、およびy_iを用いて次のように表される。

【０００６】

$A = (a_{g-1}, a_{g-2}, \ldots, a_1, a_0)$

$B = (b_{g-1}, b_{g-2}, \ldots, b_1, b_0)$

$N = (n_{g-1}, n_{g-2}, \ldots, n_1, n_0)$

$Y = (y_g, y_{g-1}, \ldots, y_1, y_0)$

2^{-kg}は剰余の法Ｎの上での2^{kg}の逆数であり、n'_0は、$R = 2^{kg}$として$R \cdot R^{-1} - N \cdot N' = 1$（$0 \leqq R^{-1} < N$，$0 \leqq N' < R$）を満たすＮ’をｋビット＝１ブロックで分割した時の最下位ブロック（０番目のブロック）を表す。また、ｃ１、ｃ２、ｍ、ｔｍｐはｋビットのワーク変数であり、ｉおよびｊはループ変数である。

【０００７】

図１２のアルゴリズムを⑥の処理を除いて回路化した場合、図１３のような回路になることが多いと考えられる。図１３の回路では、Ａ、Ｂ、およびＮがそれぞれｇ個のブロックに分割されて、ブロック毎にブロック単位演算器１０１に入力され、演算結果ｙ’$_{i}$およびｙ’$_{i-1}$がそれぞれＹのブロックｙ$_{i}$およびｙ$_{i-1}$に格納される。このとき、演算に必要なｙ$_{i}$がＹから読み出されてブロック単位演算器１０１に入力される。

【０００８】

ブロック単位演算器１０１は、図１３の下方に示すように、レジスタ１１１、１１２、１１３、１１４、１１５、１２１、１２２、１２３、１２４、１２５、１２６、セレクタ１１６、１１７、１１８、１１９、および乗算加算器１２０を備える。乗算加算器１２０はＡ×Ｂ＋Ｃ＋Ｄの演算を行う。図１３の回路と図１２のアルゴリズムの①～⑤の処理の対応関係は次の通りである。

①の処理：（ｃ１，ｔｍｐ）＝ａ$_{i}$＊ｂ$_{j}$＋ｙ$_{i}$＋ｃ１

セレクタ１１６によりＡ＝ａ$_{i}$と選択し、セレクタ１１７によりＢ＝ｂ$_{j}$と選択し、セレクタ１１８によりＣ＝ｃ１と選択し、セレクタ１１９によりＤ＝ｙ$_{i}$と選択して、乗算加算器１２０の出力の上位ｋビットをｃ１としてレジスタ１２１に格納し、下位ｋビットをｔｍｐとしてレジスタ１２６に格納する。

②の処理：ｍ＝ｔｍｐ＊ｎ’$_{0}$

セレクタ１１６によりＡ＝ｔｍｐと選択し、セレクタ１１７によりＢ＝ｎ’$_{0}$と選択し、セレクタ１１８によりＣ＝０と選択し、セレクタ１１９によりＤ＝０と選択して、乗算加算器１２０の出力の下位ｋビットをｍとしてレジスタ１２５に格納する。

③の処理：（ｃ２，ｔｍｐ）＝ｍ＊ｎ$_{i}$＋ｔｍｐ＋ｃ２

セレクタ１１６によりＡ＝ｎ$_{i}$と選択し、セレクタ１１７によりＢ＝ｍと選択し、セレクタ１１８によりＣ＝ｔｍｐと選択し、セレクタ１１９によりＤ＝ｃ２と選択して、乗算加算器１２０の出力の上位ｋビットをｃ２としてレジスタ１２２に格納し、下位ｋビットをｔｍｐとしてレジスタ１２６に格納する。

④の処理：（ｃ２，ｙ$_{i-1}$）＝ｍ＊ｎ$_{i}$＋ｔｍｐ＋ｃ２

セレクタ１１６によりＡ＝n_iと選択し、セレクタ１１７によりＢ＝ｍと選択し、セレクタ１１８によりＣ＝ｔｍｐと選択し、セレクタ１１９によりＤ＝ｃ２と選択して、乗算加算器１２０の出力の上位ｋビットをｃ２としてレジスタ１２２に格納し、下位ｋビットをy'_{i-1}としてレジスタ１２４に格納する。レジスタ１２４の内容はＹのy_{i-1}に格納される。

⑤の処理：（y_i，y_{i-1}）＝y_i＋ｃ１＋ｃ２

セレクタ１１６によりＡ＝y'_iと選択し、セレクタ１１７によりＢ＝１と選択し、セレクタ１１８によりＣ＝ｃ１と選択し、セレクタ１１９によりＤ＝ｃ２と選択して、乗算加算器１２０の出力の上位ｋビットをy'_iとしてレジスタ１２３に格納し、下位ｋビットをy'_{i-1}としてレジスタ１２４に格納する。レジスタ１２３および１２４の内容はそれぞれＹのy_iおよびy_{i-1}に格納される。

【０００９】

図１３の回路に図１２のアルゴリズムの⑥の処理を追加して、モンゴメリ乗算剰余の多倍長演算アルゴリズムの回路として完成させる場合、⑥以外の処理と同じように、ＹおよびＮをｋビット＝１ブロックで分割してブロック毎の演算を繰り返す図１４のような多倍長減算を採用して、図１５のような回路にすることが多いと考えられる。

【００１０】

図１５では、図１４の上方に示す多倍長減算を２の補数表現を用いてその下方に示すような加算に変形してから回路化している。２の補数は、コンピュータ内で負数を表現するために用いられ、１の補数に１を加えることで得られる。また、１の補数は、ビットパターンを反転させることで得られる。

【００１１】

図１５では、図１３のブロック単位演算器１０１がブロック単位演算器１３１に置き換えられている。ブロック単位演算器１３１は、ブロック単位演算器１０１に反転／非反転回路１４１を加えた構成を有する。

【００１２】

図１５の回路の動作のうち、図１２のアルゴリズムの⑥以外の処理（通常処理

）に対応する動作は図１３の回路と同じであるため、⑥の処理（最終処理）に対応する動作のみを以下に説明する。この処理では、最終処理状態信号により、反転／非反転回路１４１はn_iを反転して出力し、レジスタ１２３のy'_iは1に初期化され、ブロック単位演算器１３１から出力されたy'_{i-1}はYのy_iに格納される。

（１）y'_iを１に初期化する。

（２）次の動作を０≦ｉ≦ｇの範囲で繰り返す。

【００１３】

セレクタ１１６によりＡ＝n_iの反転（１の補数）と選択し、セレクタ１１７によりＢ＝１と選択し、セレクタ１１８によりＣ＝y'_i（$y_{i-1}-n_{i-1}$の結果からのキャリー）と選択し、セレクタ１１９によりＤ＝y_iと選択して、乗算加算器１２０の出力の上位ｋビットをy'_iとしてレジスタ１２３に格納する。

（３）y'_iのｂｉｔ０（最下位ビット）が１（Ｙ≧Ｎ）の場合は（４）へ進み、y'_iのｂｉｔ０が０（Ｙ＜Ｎ）の場合は（５）へ進む。

（４）次の動作を０≦ｉ≦ｇの範囲で繰り返す。

【００１４】

セレクタ１１６によりＡ＝n_iの反転（１の補数）と選択し、セレクタ１１７によりＢ＝１と選択し、セレクタ１１８によりＣ＝y'_i（$y_{i-1}-n_{i-1}$の結果からのキャリー）と選択し、セレクタ１１９によりＤ＝y_iと選択して、乗算加算器１２０の出力の上位ｋビットをy'_iとしてレジスタ１２３に格納し、下位ｋビットをy'_{i-1}としてレジスタ１２４に格納する。レジスタ１２４の内容はYのy_iに格納される。

（５）動作を終了する

【００１５】

【特許文献１】

特開平１１－２１２４５６号公報

【００１６】

【発明が解決しようとする課題】

しかしながら、上述した仮想的な多倍長演算回路には、次のような問題がある。

【００１７】

図１５の回路において、Ａ、Ｂ、Ｎ、Ｙ、ｃ１、ｃ２、ｍ、およびｔｍｐを格納する回路は、通常、動作クロックに同期したＲＡＭ（random access memory）もしくはＦＦ（flip-flop ）で構成される。このため、乗算加算器１２０とその直前のセレクタ１１６～１１９は、上記ＲＡＭもしくはＦＦの出力（Ａ，Ｂ，Ｎ，Ｙ，ｃ１，ｃ２，ｍ，ｔｍｐ）からＲＡＭもしくはＦＦの入力（Ｙ，ｃ１，ｃ２，ｍ，ｔｍｐ）までの通り道となるので、乗算加算器１２０とセレクタ１１６～１１９の総遅延時間が動作クロックの周期より小さくなければ、回路全体として動作しないということになる。

【００１８】

したがって、図１５の回路の動作周波数を向上させる際のボトルネックは、ブロック単位演算器１３１の最大遅延経路＜n_i→反転／非反転回路１４１→セレクタ１１６→乗算加算器１２０→（ｃ１，ｃ２，y'_i，y'_{i-1}，ｍ，ｔｍｐ）＞であり、この経路をいかに短縮するかが問題となる。

【００１９】

本発明の課題は、モンゴメリ乗算剰余の多倍長演算を行う回路においてブロック単位演算器における減算のための遅延時間を短縮し、動作周波数を維持したままで多倍長演算を行うための演算装置を提供することである。

【００２０】

【課題を解決するための手段】

図１は、本発明の演算装置の原理図である。図１の演算装置は、部分積生成手段２０１、エンコーダ手段２０２、選択手段２０３、および加算手段２０４を備え、ビットパターンで表された被乗数Ａと乗数Ｂの乗算を行う。

【００２１】

部分積生成手段２０１は、被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する。エンコーダ手段２０２は、乗数Ｂを２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i}

＋１、ｂ２ｉ、およびｂ２ｉ－１を指定するｉの値に応じた選択信号を出力する。選択手段２０３は、選択信号に応じて、複数の部分積のいずれかを選択して出力する。加算手段２０４は、選択手段２０３から出力される、ｉの数と同じ数だけの部分積を加算して、乗算結果を生成する。

【００２２】

そして、エンコーダ手段２０２が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、加算手段２０４が、－Ａを表す部分積から被乗数Ａの２の補数を生成して、被乗数Ａの２の補数を乗算結果として出力するような動作モードを、演算装置に設ける。この動作モードにおいて、加算手段２０４が、－Ａを表す部分積から被乗数Ａの２の補数を生成する代わりに、被乗数Ａの１の補数を生成するようにすることもできる。

【００２３】

２次Ｂｏｏｔｈアルゴリズムは、ＡおよびＢが２の補数の整数である場合の乗算Ａ×Ｂにおいて部分積の個数を削減するためのアルゴリズムとして知られている。部分積生成手段２０１は、被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積として、Ａ、２Ａ、－Ａ、－２Ａ、および０を表す部分積を生成する。

【００２４】

上記動作モード（後述する補数モード）において、エンコーダ手段２０２は、乗数Ｂの連続する３つのビットｂ２ｉ＋１、ｂ２ｉ、およびｂ２ｉ－１をエンコードして、ｂ１、ｂ０、およびｂ－１の３つのビットに対しては－Ａを表す部分積を選択するための選択信号を生成し、ｂ２ｉ＋１、ｂ２ｉ、およびｂ２ｉ－１（ｉ≠０）の３つのビットに対しては０を表す部分積を選択するための選択信号を生成する。

【００２５】

これにより、選択手段２０３は、ｂ１、ｂ０、およびｂ－１の３つのビットに対しては－Ａを表す部分積を選択し、それ以外のビットに対しては０を表す部分積を選択するので、加算手段２０４は、－Ａを表す部分積と０とを加算すること

になる。－ＡはＡの２の補数またはＡの１の補数で表されるから、この動作モードにおける乗算結果は、Ａの２の補数またはＡの１の補数（＝－Ａ）となる。

【００２６】

これに対して、通常の動作モードにおいては、演算装置は乗算Ａ×Ｂの結果を出力することができる。このように、乗算Ａ×Ｂを行う演算装置において－Ａを乗算結果として出力する動作モードを設けることで、図１５の反転／非反転回路１４１がなくても図１２の⑥の処理を行うことが可能となる。この演算装置をブロック単位演算器１３１に組み込めば、ブロック単位演算器１３１における遅延時間を短縮して、動作周波数を向上させることができる。

【００２７】

図１の部分積生成手段２０１は、例えば、後述する図５のシフト回路２４１、２４３、および反転回路２４２に対応し、図１のエンコーダ手段２０２は、例えば、図５のエンコーダ２４７に対応する。また、図１の選択手段２０３は、例えば、図５の選択回路２４４に対応し、図１の加算手段２０４は、例えば、図５の多段加算回路２４５に対応する。

【００２８】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

本実施形態では、乗算加算器のアーキテクチャに２次Ｂｏｏｔｈアルゴリズムを採用して、図１５の回路のブロック単位演算器１３１の最大遅延経路＜n_i→反転／非反転回路１４１→セレクタ１１６→乗算加算器１２０→（ｃ１，ｃ２，y'_i，y'_{i-1}，ｍ，ｔｍｐ）＞上の反転／非反転回路１４１を乗算加算器１２０に吸収させる。これにより、ブロック単位演算器１３１における遅延時間を短縮して、動作周波数を向上させる。

【００２９】

また、図１５の回路では、反転／非反転回路１４１の規模がブロックビット数ｋに比例するため、ｋが大きくなるほど、図１３の回路の規模に対する増分が顕著になる。しかし、本実施形態では、反転／非反転回路１４１を乗算加算器１２０に吸収することにより、図１３の回路とほぼ同じ規模で図１２のアルゴリズム

を実現することができる。

【００３０】

２次Ｂｏｏｔｈアルゴリズムは、ＡおよびＢが$-2^{q-1} \leqq A, B \leqq 2^{q-1}-1$でｑビットの２の補数の整数である場合の乗算Ａ×Ｂについて、ＡおよびＢのビット表現をそれぞれ$\{a_{q-1}, a_{q-2}, \ldots, a_1, a_0\}$および$\{b_{q-1}, b_{q-2}, \ldots, b_1, b_0\}$として、以下の手順により乗算Ａ×Ｂの部分積の個数をｑ／２に削減するアルゴリズムである。

・Ｂの多項式表現である式（１）を式（２）に変形する。

・さらに、d_iを－２、－１、０、＋１、および＋２を取りうるビットと定義して、式（２）を式（３）で表す。

式（１）　$-b_{q-1}2^{q-1}+b_{q-2}2^{q-2}+\ldots$
$+b_1 2^1+b_0 2^0$

式（２）　$\Sigma\left((-2b_{2i+1}+b_{2i}+b_{2i-1})2^{2i}\right)$
※$0 \leqq i \leqq q/2-1$，$b_{-1}=0$

式（３）　$\Sigma d_i 2^{2i}$　（※$d_i=-2b_{2i+1}+b_{2i}+b_{2i-1}$）
※$0 \leqq i \leqq q/2-1$，$b_{-1}=0$

ｑは偶数である必要があるが、ｑが奇数であっても＋１補正して２次Ｂｏｏｔｈアルゴリズムを適用することができる。ＡおよびＢが$0 \leqq A, B \leqq 2^k-1$のｋビットの整数である場合の乗算Ａ×Ｂについても、ｋが偶数であれば$q=k+2$、$a_{q-1}=a_{q-2}=b_{q-1}=b_{q-2}$として、ｋが奇数であれば$q=k+1$、$a_{q-1}=b_{q-1}$として２次Ｂｏｏｔｈアルゴリズムを適用することができる。

【００３１】

図２は、ＡおよびＢが$0 \leqq A, B \leqq 2^k-1$のｋビットの整数である場合の２次Ｂｏｏｔｈアルゴリズムによる乗算Ａ×Ｂを示している。この場合の動作は次

の通りである。

①ｋビットのＡおよびＢをｑビットに補正してＡ’およびＢ’とする。このとき、ｋが偶数であればｑ＝ｋ＋２、$a_{q-1}=a_{q-2}=b_{q-1}=b_{q-2}$とし、ｋが奇数であればｑ＝ｋ＋１、$a_{q-1}=b_{q-1}$とする。

②$0 \leqq i \leqq q/2-1$とした場合のＢ’の３つのビットb_{2i+1}、b_{2i}、およびb_{2i-1}を、Ｂｏｏｔｈエンコーダ２１１を用いてd_iにエンコードする。

③b_{2i+1}、b_{2i}、およびb_{2i-1}から得られたd_iを－２、－１、０、＋１、および＋２を取りうるＢ’のビットとして用いて、乗算Ａ’×Ｂ’を行う。この乗算における各部分積はＡ’×d_i（２の補数）である。

④積Ａ’×Ｂ’（２ｑビット）から下位２ｋビットを切り出し、積Ａ×Ｂ（２ｋビット）とする。

【００３２】

図３は、２次Ｂｏｏｔｈアルゴリズムを採用した乗算器に、出力が－Ａ（２の補数）になるモード（補数モード）を追加した乗算器を示している。図３の乗算器においては、モード選択信号により、通常の乗算Ａ×Ｂを行うモード（通常モード）と補数モードとが切り替えられる。

【００３３】

補数モード時には、Ｂｏｏｔｈエンコーダ２２１の出力d_0が－１になり、それ以外の出力d_i（$i \neq 0$）が０になるため、部分積Ａ’×d_0は－Ａ’（２の補数）になり、他の部分積Ａ’×d_i（$i \neq 0$）は０になる。したがって、乗算器の出力Ｘは－Ａ（２の補数）になる。補数モード時のＢはｄｏｎ’ｔ　ｃａｒｅで構わない。

【００３４】

ここでは、部分積Ａ’×d_0を２の補数としている。２の補数は１の補数に１を加えたものであるから、２の補数を回路で実現するときは、ビットパターンを１の補数にして（つまり、ビット反転して）から、＋１加算する。しかし、この操作をそのまま行うと、部分積加算にとても時間がかかる。

【００３５】

そこで、部分積Ａ’×ｄ$_0$を１の補数としておき、２の補数にするための＋１加算を部分積加算と一緒に行うのが一般的である。図３のＢｏｏｔｈエンコーダ２２１のｓ$_i$は、この＋１加算を実現するための補正値として用いられ、ｄ$_i$＜０のときｓ$_i$＝１となり、ｄ$_i$≧０のときｓ$_i$＝０となる。

【００３６】

図３の乗算器では、補数モード時のＢがｄｏｎ’ｔ　ｃａｒｅで構わないのに対して、補数モード時にＢ＝０を入力すれば、補数モード時のｄ$_i$（ｉ≠０）として通常モード時と同じ値を用いることができる。

【００３７】

図４は、このような乗算器を示している。補数モード時には、Ｂｏｏｔｈエンコーダ２３１の出力ｄ$_0$が－１になり、それ以外の出力ｄ$_i$（ｉ≠０）は、Ｂ＝０の場合の通常モード時の出力ｄ$_i$（ｉ≠０）と同じ０になる。このため、部分積Ａ’×ｄ$_0$は－Ａ’（２の補数）になり、他の部分積Ａ’×ｄ$_i$（ｉ≠０）は０になる。したがって、乗算器の出力Ｘは－Ａ（２の補数）になる。

【００３８】

図４の乗算器は、図３の乗算器に比べてＢｏｏｔｈエンコーダの規模が小さくて済むため、乗算器を使用する側でＢ＝０を入力するのが容易である場合は、最適な乗算器であるといえる。

【００３９】

図３および図４の乗算器は、例えば、図５に示すような回路で構成される。図５の回路は、レジスタ２４０、２４６、シフト回路２４１、２４３、反転回路２４２、選択回路２４４、多段加算回路２４５、およびエンコーダ２４７を備える。エンコーダ２４７は、図３のＢｏｏｔｈエンコーダ２２１または図４のＢｏｏｔｈエンコーダ２３１に対応する。

【００４０】

レジスタ２４６のＢ’は２ビットずつの組に分割され、各組のビット対が下位の組の上位ビットとともにエンコーダ２４７に順番に入力される。エンコーダ２４７は、これらの３つの入力ｂ２ｉ＋１、ｂ２ｉ、およびｂ２ｉ－１からｄ$_i$に相当する選択信号を生成し、Ａ’の倍数を選択する選択回路２４４に出力する。

【００４１】

シフト回路２４１、２４３は、入力されたビットパターンを１ビットシフトし、反転回路２４２は、入力されたビットパターンを反転させる。これらの回路により、レジスタ２４０のA’からは、直接（A’）、１ビットシフト（２A’）、反転（－A’（１の補数））、および反転の１ビットシフト（－２A’）の４種類の値が生成され、選択回路２４４に入力される。

【００４２】

選択回路２４４は、エンコーダ２４７の各ビット対出力に対応して設けられており、エンコーダ２４７からの選択信号に応じて４種類の入力値の１つを選択し、部分積A’×d_iとして多段加算回路２４５に出力する。このとき、選択回路２４４からは補正値s_iも同時に出力される。多段加算回路２４５は、入力された部分積A’×d_iおよび補正値s_iを加算して、乗算結果X（＝A×B）を出力する。

【００４３】

図３および図４の乗算器では、補数モード時にd_0＝－１およびs_0＝１とすることで、出力を－A（２の補数）としているが、s_0＝０とすればこの出力を－A（１の補数）にすることができる。図１４に示したように、多倍長減算では、ビットパターン全体として２の補数であってもブロック単位の演算では１の補数を扱うことになるので、乗算器の出力を－A（１の補数）にすることが重要である。この場合、部分積A’×d_0に対応する－A’を生成する回路は、図６に示すように、反転回路２５１および加算回路２５２を用いて構成される。

【００４４】

図６の反転回路２５１は、A’を反転して出力し、加算回路２５２は、反転されたA’と補正値２５３とを加算して出力する。補正値２５３は、通常モード時に１となり、補数モード時に０となるので、通常モード時は－A’（２の補数）が出力され、補数モード時は－A’（１の補数）が出力される。

【００４５】

図６の反転回路２５１および加算回路２５２は、それぞれ、図５の反転回路２４２および多段加算回路２４５に対応し、補正値２５３は補正値s_0に対応する

。この構成では、通常モードと補数モードとでs_0が異なる値をとることになる。

【００４６】

図５の乗算器の出力と、$0 \leqq C \leqq 2^{k-1}-1$および$0 \leqq D \leqq 2^{k-1}-1$であるｋビットのＣおよびＤとを加算する、Ａ×Ｂ＋Ｃ＋Ｄの乗算加算器は、図７のような回路で構成される。図７の回路では、図５の多段加算回路２４５の代わりに、多段加算回路２６１が用いられている。

【００４７】

多段加算回路２６１は、選択回路２４４からの部分積および補正値に加えて、外部から入力されるＣおよびＤを加算し、乗算加算結果をＸ（＝Ａ×Ｂ＋Ｃ＋Ｄ）を出力する。補数モード時は、Ｘ＝－Ａ（２の補数）＋Ｃ＋Ｄとなる。

【００４８】

しかし、図６の－Ａ’生成回路のような構成を採用すれば、補数モード時の出力は、Ｘ＝－Ａ（１の補数）＋Ｃ＋Ｄとなる。この場合、図６の反転回路２５１および加算回路２５２は、それぞれ、図７の反転回路２４２および多段加算回路２６１に対応し、補正値２５３は補正値s_0に対応する。

【００４９】

図８および図９は、図７において補数モード時の出力を－Ａ（１の補数）＋Ｃ＋Ｄとした乗算加算器を使用して、図１４の多倍長減算（Ｙ＝Ｙ－Ｎ）を実行する回路を示している。図８の多倍長減算回路は、レジスタ２７１、２７２、２７３、２７４、２７５、２７７、乗算加算器２７６、および反転回路２７８を備え、図９の多倍長減算回路は、図８において乗算加算器２７６の代わりに乗算加算器２８１を用いた構成を有する。

【００５０】

図８の乗算加算器２７６は、図３のＢｏｏｔｈエンコーダ２２１に従って乗算を行うため、入力Ｂはｄｏｎ’ｔ　ｃａｒｅとなっている。これに対して、図９の乗算加算器２８１は、図４のＢｏｏｔｈエンコーダ２３１に従って乗算を行うため、入力Ｂは０となっている。図８および図９の回路の動作は次の通りである。

（０）レジスタ２７７のキャリー（ｃａｒｒｙ）の値を１に初期化する。

（１）レジスタ２７１のＮからn_iを選択してレジスタ２７３に格納し、レジスタ２７２のＹからy_iを選択してレジスタ２７４に格納する。

（２）乗算加算器２７６は、入力をＡ＝n_i、Ｂ＝ｄｏｎ’ｔ　ｃａｒｅ、Ｃ［ｋ－１：１］＝０、Ｃ［０］＝ｃａｒｒｙ（ｉ－１番目のブロックの演算結果からのキャリー）、Ｄ＝y_iとして、補数モード時のｋビットの演算Ｘ＝－Ａ（１の補数）＋Ｃ＋Ｄを実行する。

【００５１】

また、乗算加算器２８１は、入力をＡ＝n_i、Ｂ＝０、Ｃ［ｋ－１：１］＝０、Ｃ［０］＝ｃａｒｒｙ、Ｄ＝y_iとして、Ｘ＝－Ａ（１の補数）＋Ｃ＋Ｄを実行する。

（３）乗算加算器２７６および２８１の出力のうち、Ｘ［ｋ－１：０］をy'_{i-1}としてレジスタ２７５に保存し、Ｘ［ｋ］を反転回路２７８により反転させてレジスタ２７７に格納する。レジスタ２７７に格納されたｃａｒｒｙは、ｉ＋１番目のブロックへのキャリーとなる。ここで、Ｘ［ｋ］を反転させるのは、図１４に示したように、補数のキャリーが必要となるためである。Ｘ［２ｋ－１：ｋ＋１］は不要なので、無視される。

（４）レジスタ２７５のy'_{i-1}をy_iとしてレジスタ２７２に格納する。

（５）（１）～（４）の動作を０≦ｉ≦ｇの範囲で繰り返す。

（６）（５）の動作が終了した時点で、レジスタ２７２のＹにはＹ－Ｎの演算結果が格納されている。また、ｇ番目のブロックの演算結果Ｘ［ｋ］は、Ｙ＝Ｙ－Ｎの符号（１：Ｙ≧Ｎ，０：Ｙ＜Ｎ）を表す。

【００５２】

図１０および図１１は、図７において補数モード時の出力を－Ａ（１の補数）＋Ｃ＋Ｄとした乗算加算器を使用して、モンゴメリ乗算剰余の多倍長演算（Ｙ＝ＡＢ２$^{-k}$ｇ　ｍｏｄ　Ｎ）を実行する回路を示している。図１０の多倍長演算回路は、レジスタ２９１、２９２、２９５、２９９、ブロック選択回路２９３、２９４、２９６、２９８、３００、およびブロック単位演算器２９７を備え、図１１の多倍長演算回路は、図１０においてブロック単位演算器２９７の代わりに

ブロック単位演算器３３１を用いた構成を有する。

【００５３】

図１０において、レジスタ２９１、２９２、および２９５には、Ａ、Ｂ、およびＮがそれぞれｇ個のブロックに分割されて格納され、ブロック選択回路２９３、２９４、および２９６は、それぞれＡ、Ｂ、およびＮのブロックa_i、b_i、およびn_iを選択して、ブロック単位演算器２９７に出力する。また、ブロック選択回路２９８は、レジスタ２９９からＹのブロックy_iを選択して、ブロック単位演算器２９７に出力する。

【００５４】

ブロック単位演算器２９７は、ブロックa_i、b_i、n_i、n'_0、およびy'_iを入力として演算を行い、演算結果y'_iおよびy'_{i-1}を出力する。ブロック選択回路３００は、y'_iおよびy'_{i-1}のいずれかを選択し、ブロックy_iまたはy_{i-1}としてレジスタ２９９に格納する。図１１の多倍長演算回路の動作も同様である。

【００５５】

ブロック単位演算器２９７は、図１０の下方に示すように、レジスタ３１１、３１２、３１３、３１４、３１５、３２１、３２２、３２３、３２４、３２５、３２６、セレクタ３１６、３１７、３１８、３１９、乗算加算器２７６、および反転回路３２０を備える。図１１のブロック単位演算器３３１は、図１０において乗算加算器２７６の代わりに乗算加算器２８１を用い、さらにセレクタ３３２を追加した構成を有する。

【００５６】

図１０および図１１の回路では、図１２のアルゴリズムの⑥以外の処理（通常処理）と⑥の処理（最終処理）が、処理状態信号３０１によって切り替えられる。乗算加算器２７６および２８１は、通常処理のときは通常モードの演算（Ｘ＝Ａ×Ｂ＋Ｃ＋Ｄ）を行い、最終処理のときは補数モードの演算（Ｘ＝－Ａ（１の補数）＋Ｃ＋Ｄ）を行う。また、図１１のセレクタ３３２は、通常処理のときは入力１を選択し、最終処理のときは入力０を選択して、乗算加算器２８１に出力する。

【００５７】

図１０および図１１の回路の通常処理に対応する動作は図１３の回路と同じであるため、最終処理に対応する動作のみを説明することにする。図１０および図１１の回路の処理⑥に相当する動作は次の通りである。

（０）レジスタ３２３のy’$_i$の値を１に初期化し、乗算加算器２７６および２８１を補数モードに設定する。

（１）レジスタ２９５のNからn_iを選択してレジスタ３１２に格納し、レジスタ２９９のYからy_iを選択してレジスタ３１５に格納する。

（２）セレクタ３１６、３１８、および３１９は、それぞれ、A＝n_i、C＝y’$_i$のｂｉｔ０を反転回路３２０により反転した値（ｉ－１番目のブロックの演算結果からのキャリー）、およびD＝y_iと選択する。

【００５８】

このとき、図１０の回路では、セレクタ３１７の出力Bはｄｏｎ’ｔ　ｃａｒｅで構わない。これに対して、図１１の回路では、セレクタ３３２が入力０を選択し、セレクタ３１７がセレクタ３３２の出力を選択するので、結局、B＝０と選択される。

【００５９】

そして、乗算加算器２７６および２８１は、補数モード時のｋビットの演算X＝－A（１の補数）＋C＋Dを実行する。

（３）乗算加算器２７６および２８１の出力のうち、X［ｋ－１：０］をy’$_i$$_{-1}$としてレジスタ３２４に保存し、X［２ｋ－１：ｋ］をy’$_i$としてレジスタ３２３に格納する。レジスタ３２３に格納されたy’$_i$のｂｉｔ０（X［ｋ］）は、反転回路３２０により反転されて、ｉ＋１番目のブロックへのキャリーとなる。

（４）レジスタ３２４のy’$_{i-1}$をy_iとしてレジスタ２９９に格納する。レジスタ３２３のy’$_i$のｂｉｔ０以外のビットは不要なので、無視される。

（５）（１）～（４）の動作を０≦ｉ≦ｇの範囲で繰り返す。

（６）（５）の動作が終了した時点で、レジスタ２９９のYにはY－Nの演算結果が格納されている。また、レジスタ３２３に格納されたy’ｇのｂｉｔ０は、

Ｙ＝Ｙ－Ｎの符号（１：Ｙ≧Ｎ，０：Ｙ＜Ｎ）を表す。

【００６０】

（付記１）　ビットパターンで表された被乗数Ａと乗数Ｂの乗算を行う演算装置であって、

被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2i}、およびb_{2i-1}を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、ｉの数と同じ数だけの部分積を加算して、乗算結果を生成する加算手段とを備え、

前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの２の補数を生成して、該被乗数Ａの２の補数を前記乗算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【００６１】

（付記２）　ビットパターンで表された被乗数Ａと乗数Ｂの乗算を行う演算装置であって、

被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2i}、およびb_{2i-1}を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、ｉの数と同じ数だけの部分積を加算して、乗算

結果を生成する加算手段とを備え、

前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの１の補数を生成して、該被乗数Ａの１の補数を前記乗算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【００６２】

（付記３）　前記エンコーダ手段は、前記動作モードにおいて、乗数Ｂの値にかかわらず、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力することを特徴とする付記２記載の演算装置。

【００６３】

（付記４）　前記エンコーダ手段は、前記動作モードにおいて、乗数Ｂとして０が入力された場合に、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力することを特徴とする付記２記載の演算装置。

【００６４】

（付記５）　ビットパターンで表された被乗数Ａと乗数Ｂを乗算し、乗算結果とビットパターンで表された数Ｃおよび数Ｄとを加算する、乗算加算を行う演算装置であって、

被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2i}、およびb_{2i-1}を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、ｉの数と同じ数だけの部分積と数Ｃと数Ｄとを加算して、乗算加算結果を生成する加算手段とを備え、

前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択

信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの１の補数を生成し、該被乗数Ａの１の補数と数Ｃと数Ｄとを加算した結果を前記乗算加算結果として出力するような動作モードを設けたことを特徴とする演算装置。

【００６５】

（付記６）　ｋビットのビットパターンで表された被乗数Ａと乗数Ｂを乗算し、乗算結果とｋビットのビットパターンで表された数Ｃおよび数Ｄとを加算することで、１ブロックがｋビットであるｇ＋１個のブロックからなる整数Ｙから１ブロックがｋビットであるｇ個のブロックからなる整数Ｎを減算する演算を行う演算装置であって、

被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2i}、およびb_{2i-1}を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する選択手段と、

前記選択手段から出力される、ｉの数と同じ数だけの部分積と数Ｃと数Ｄとを加算して、２ｋビットの乗算加算結果を生成する加算手段と、

前記乗算加算結果の一部のビットを反転する反転手段とを備え、

前記部分積生成手段が、整数Ｎのｊ番目のブロックn_jを被乗数Ａとして用い、前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの１の補数を生成し、ｊ－１番目のブロックの乗算加算からのキャリーを数Ｃとして用い、整数Ｙのｊ番目のブロックy_jを数Ｄとして用いて、該被乗数Ａの１の補数と数Ｃと数Ｄとを加算した結果をｊ番目のブロックの乗算加算結果として出力し、前記反転手段が、該ｊ番目のブロックの乗算加算結果の一部のビットを反転してｊ＋１番目のブロックの乗算加算へのキャリーを生成することを特徴とする演算装置。

【００６６】

（付記７）　ビットパターンで表された整数ＩおよびＪと剰余の法Ｎを１ブロックがｋビットであるｇ個のブロックにそれぞれ分割して、Ｙ＝ＩＪ２－ｋｇ　ｍｏｄ　Ｎとなるモンゴメリ乗算剰余の多倍長演算を行う演算装置であって、

ｋビットの被乗数Ａ、乗数Ｂ、数Ｃ、および数Ｄのそれぞれについて、与えられた複数の値のいずれかを選択して出力する第１の選択手段と、

前記第１の選択手段から出力される被乗数Ａから２次Ｂｏｏｔｈアルゴリズムにおける複数の部分積を生成する部分積生成手段と、

前記第１の選択手段から出力される乗数Ｂを前記２次Ｂｏｏｔｈアルゴリズムによりエンコードして、乗数Ｂの連続する３つのビットであるb_{2i+1}、b_{2i}、およびb_{2i-1}を指定するｉの値に応じた選択信号を出力するエンコーダ手段と、

前記選択信号に応じて、前記複数の部分積のいずれかを選択して出力する第２の選択手段と、

前記第２の選択手段から出力される、ｉの数と同じ数だけの部分積と、前記第１の選択手段から出力される数Ｃと数Ｄとを加算して、２ｋビットの乗算加算結果を生成する加算手段と、

前記乗算加算結果の一部のビットを反転する反転手段とを備え、

前記第１の選択手段が、整数Ｎのｊ番目のブロックn_jを被乗数Ａとして選択し、ｊ－１番目のブロックの乗算加算からのキャリーを数Ｃとして選択し、Ｙのｊ番目のブロックy_jを数Ｄとして選択し、前記エンコーダ手段が、ｉが０のときには－Ａを表す部分積を選択するための選択信号を出力し、ｉが０以外のときには０を表す部分積を選択するための選択信号を出力し、前記加算手段が、該－Ａを表す部分積から被乗数Ａの１の補数を生成し、該被乗数Ａの１の補数と数Ｃと数Ｄとを加算した結果をｊ番目のブロックの乗算加算結果として出力し、前記反転手段が、該ｊ番目のブロックの乗算加算結果の一部のビットを反転してｊ＋１番目のブロックの乗算加算へのキャリーを生成するような動作モードを設けたことを特徴とする演算装置。

【００６７】

【発明の効果】

本発明によれば、モンゴメリ乗算剰余の多倍長減算を行うために必要となる回路を乗算加算器に吸収させることができ、モンゴメリ乗算剰余の多倍長演算アルゴリズムを、従来の回路より小さい規模のハードウェアで実現することが可能となる。

【００６８】

また、多倍長減算を行うために必要となる回路を乗算加算器に吸収させることで、回路の遅延時間が短縮されるため、従来の回路より高い動作周波数で動作させることができる。

【００６９】

モンゴメリ乗算剰余の多倍長演算アルゴリズムは、ＲＳＡ暗号や楕円曲線暗号で使われる演算の中では計算量が多いものであり、本発明はこれらの暗号処理の速度向上に寄与するところが大きい。

【図面の簡単な説明】

【図１】

本発明の演算装置の原理図である。

【図２】

２次Ｂｏｏｔｈアルゴリズムによる演算を示す図である。

【図３】

第１の乗算器を示す図である。

【図４】

第２の乗算器を示す図である。

【図５】

乗算器の回路構成図である。

【図６】

－Ａ’生成回路を示す図である。

【図７】

乗算加算器の回路構成図である。

【図８】

第１の多倍長減算回路を示す図である。

【図９】

第２の多倍長減算回路を示す図である。

【図１０】

第１の多倍長演算回路を示す図である。

【図１１】

第２の多倍長演算回路を示す図である。

【図１２】

モンゴメリ乗算剰余の多倍長演算アルゴリズムを示す図である。

【図１３】

仮想的な乗算剰余演算回路を示す図である。

【図１４】

多倍長減算を示す図である。

【図１５】

仮想的な多倍長演算回路を示す図である。

【符号の説明】

１０１、１３１、２９７、３３１　ブロック単位演算器

１１１、１１２、１１３、１１４、１１５、１２１、１２２、１２３、１２４、１２５、１２６、２４０、２４６、２７１、２７２、２７３、２７４、２７５、２７７、２９１、２９２、２９５、２９９、３１１、３１２、３１３、３１４、３１５、３２１、３２２、３２３、３２４、３２５、３２６、　レジスタ

１１６、１１７、１１８、１１９、３１６、３１７、３１８、３１９、３３２　セレクタ

１４１　反転／非反転回路

１２０、２７６、２８１　乗算加算器

２０１　部分積生成手段

２０２　エンコーダ手段

２０３　選択手段

２０４　加算手段

２１１、２２１、２３１　Ｂｏｏｔｈエンコーダ

２４１、２４３　シフト回路

２４２、２５１、２７８、３２０　反転回路

２４４　選択回路

２４５、２６１　多段加算回路

２４７　エンコーダ

２５２　加算回路

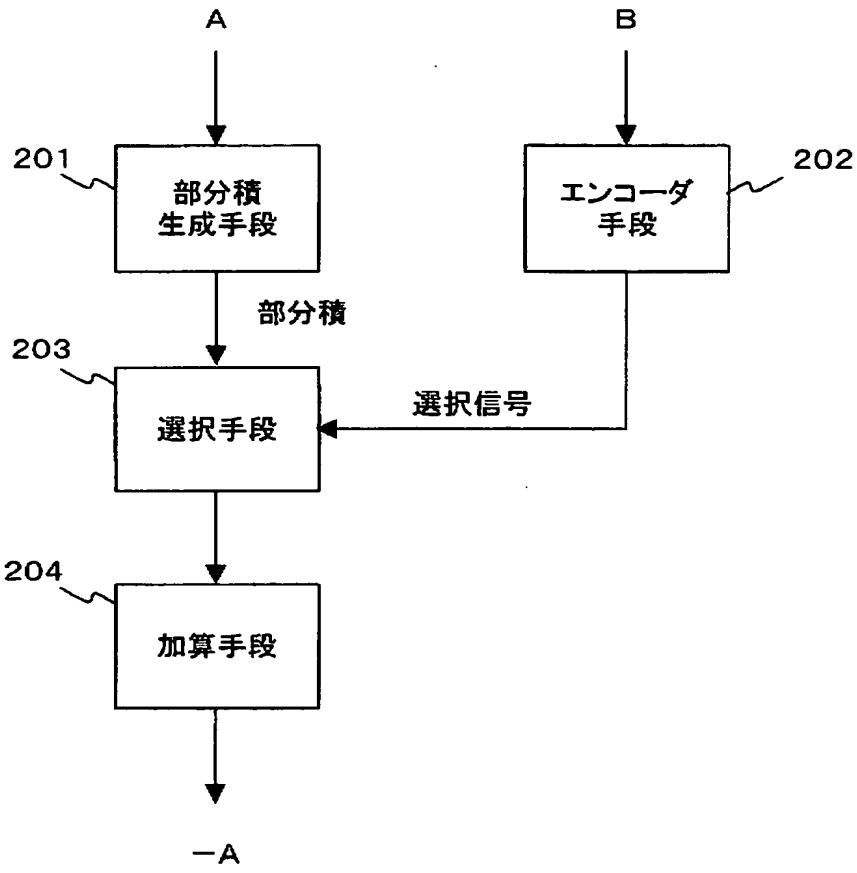
２５３　補正値

２９３、２９４、２９６、２９８、３００　ブロック選択回路

【書類名】　図面

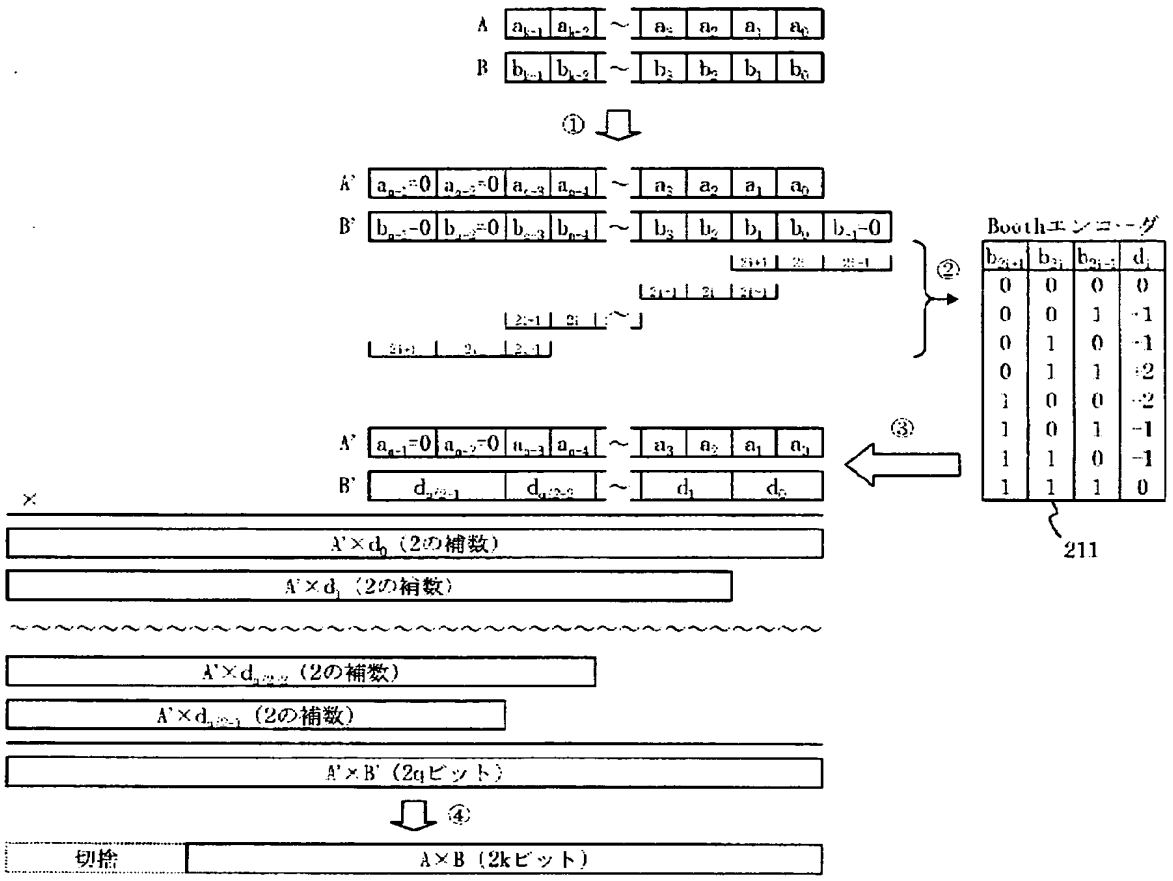
【図１】

本発明の演算装置の原理図



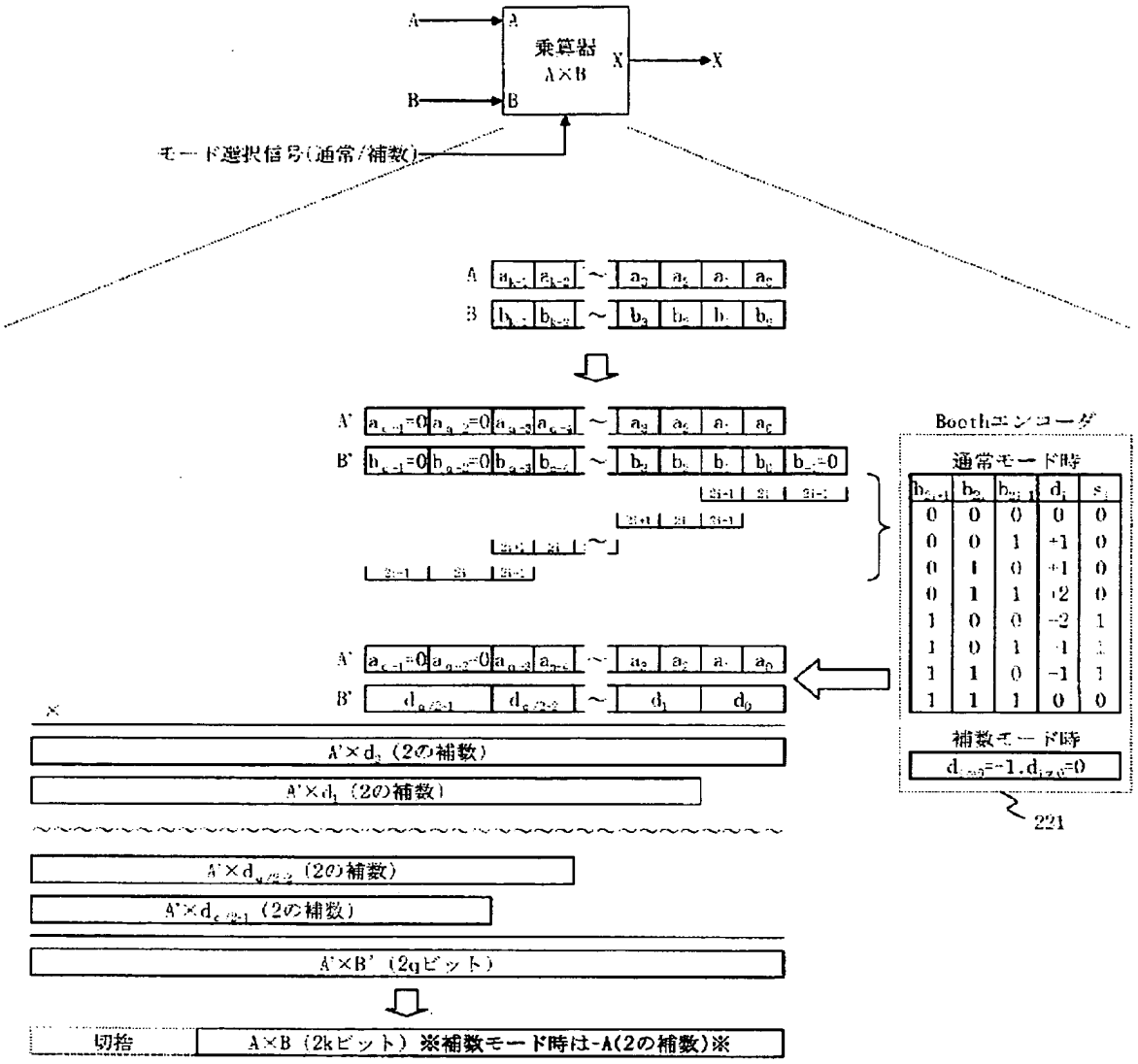
【図２】

２次Boothアルゴリズムによる乗算を示す図



【図３】

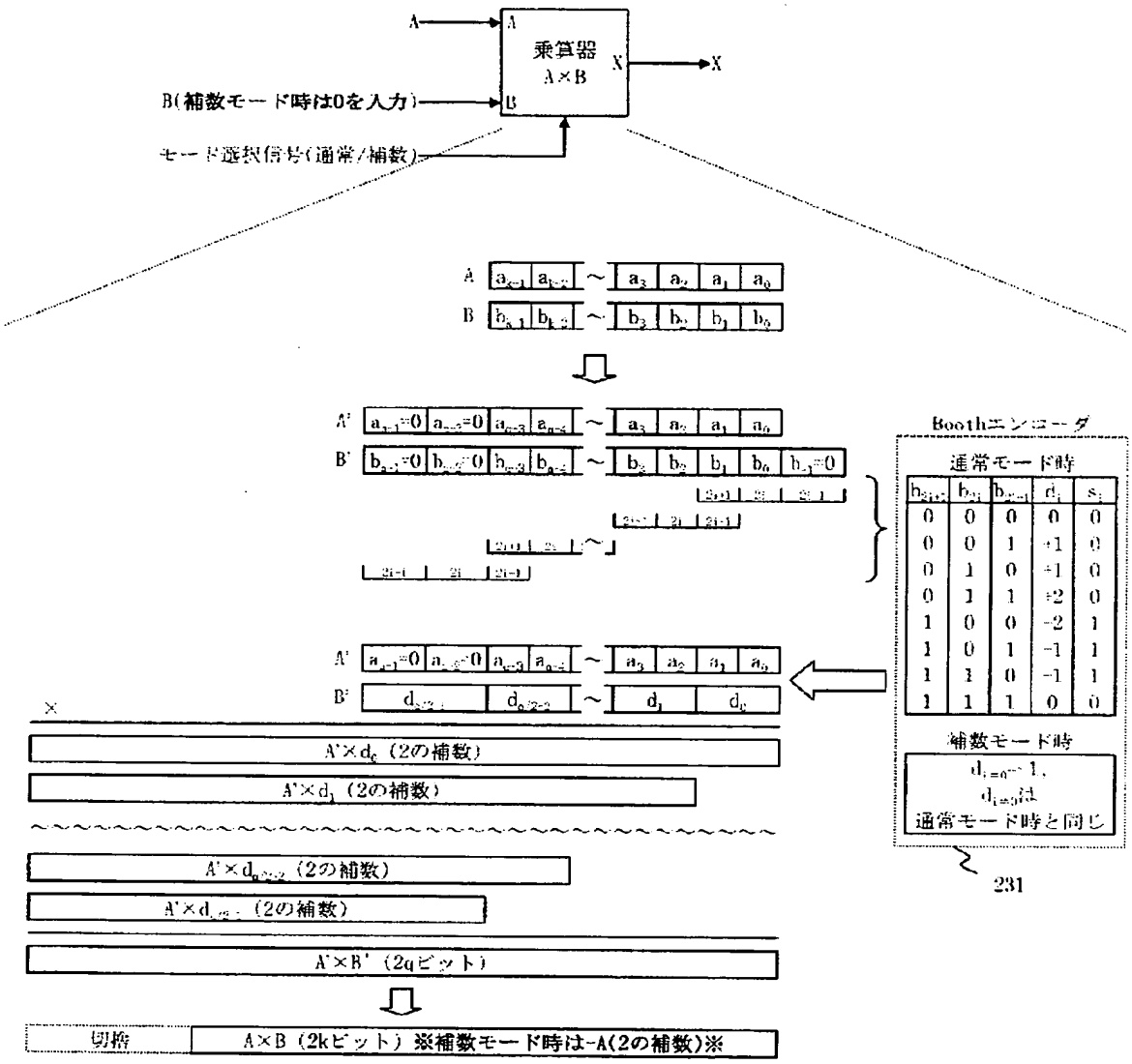
第1の乗算器を示す図



b_{2i-1}	b_{2i}	b_{2i+1}	d_i	s_i
0	0	0	0	0
0	0	1	+1	0
0	1	0	+1	0
0	1	1	+2	0
1	0	0	−2	1
1	0	1	−1	1
1	1	0	−1	1
1	1	1	0	0

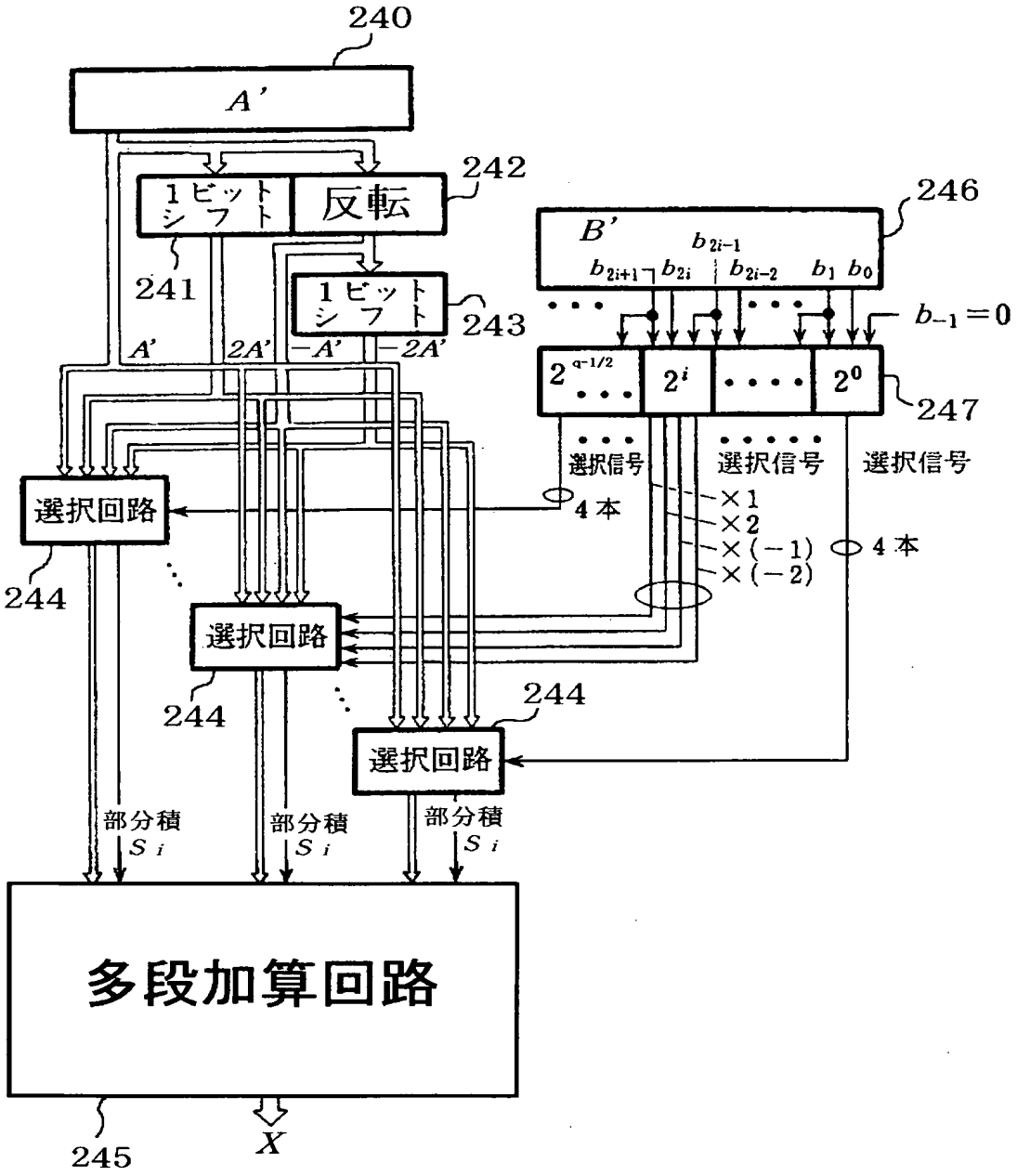
【図４】

第2の乗算器を示す図



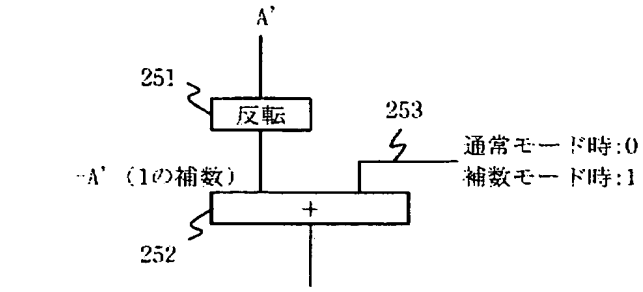
【図５】

乗算器の回路構成図



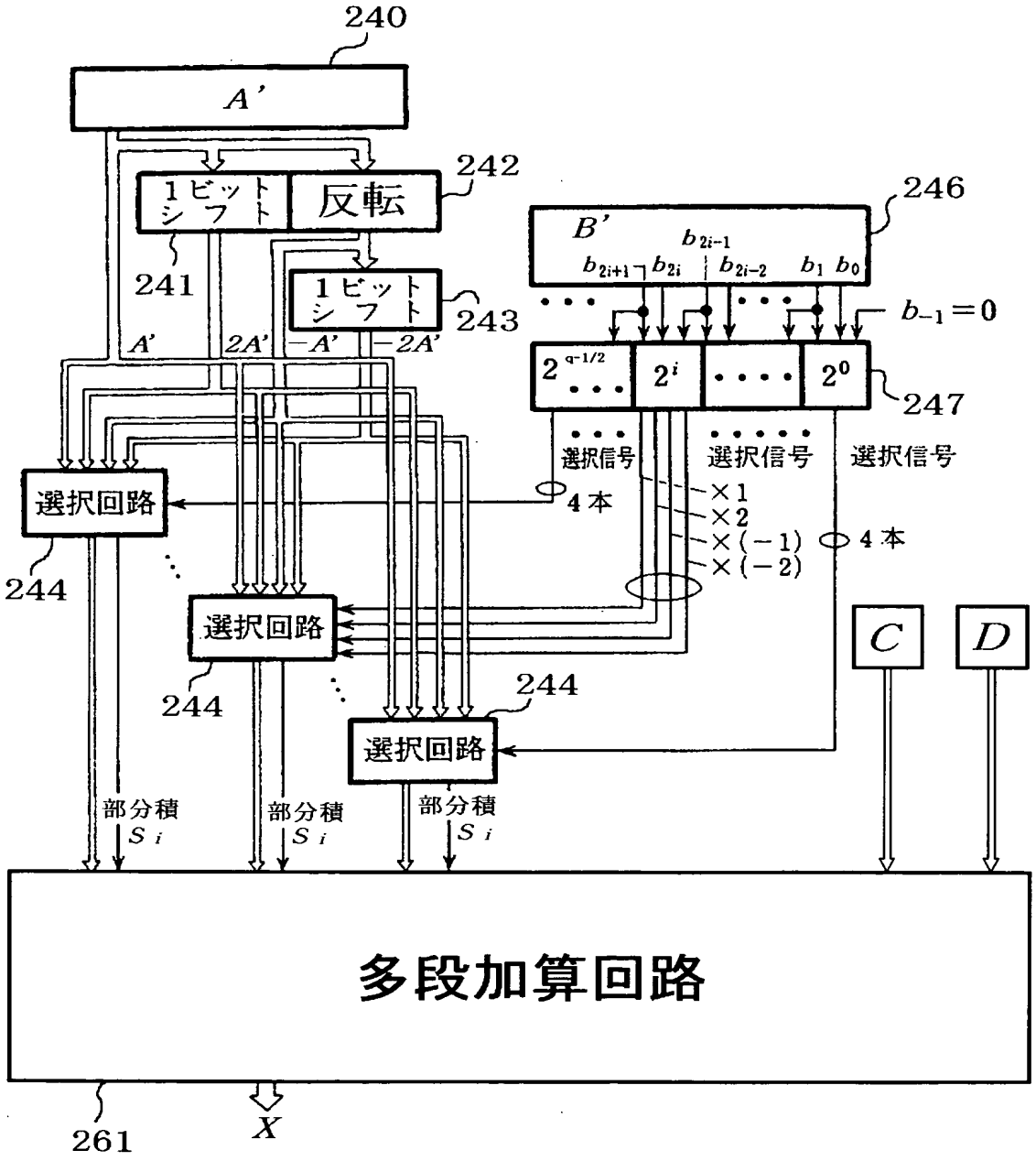
【図６】

－A’生成回路を示す図



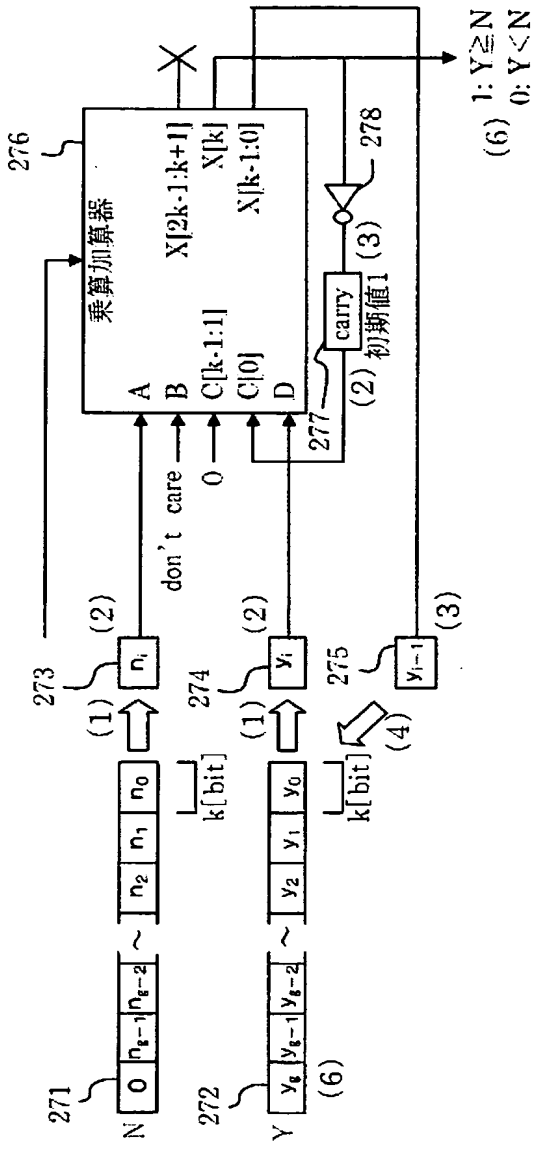
【図７】

乗算加算器の回路構成図



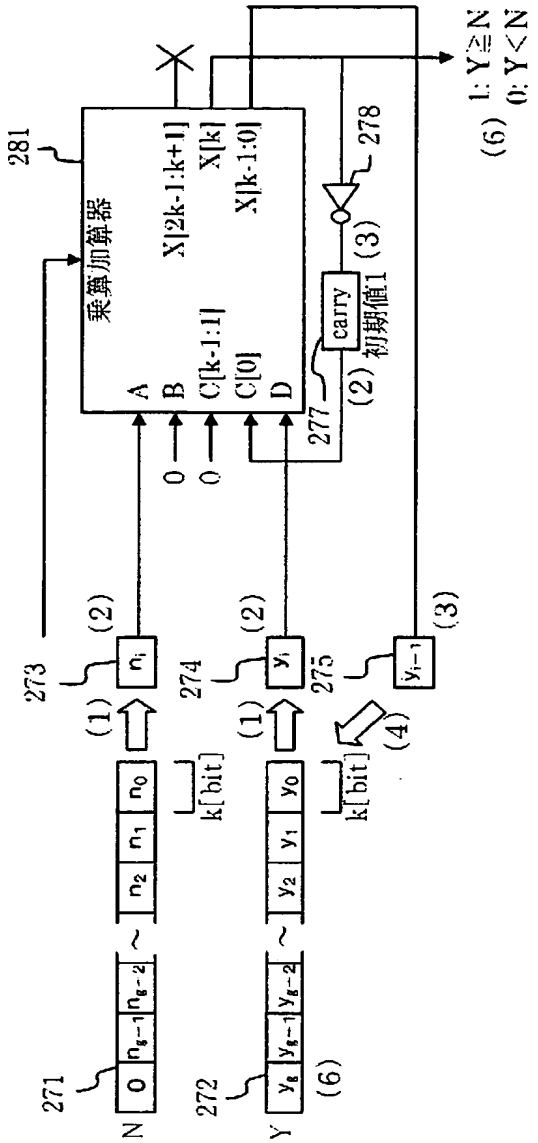
【図８】

第1の多倍長減算回路を示す図



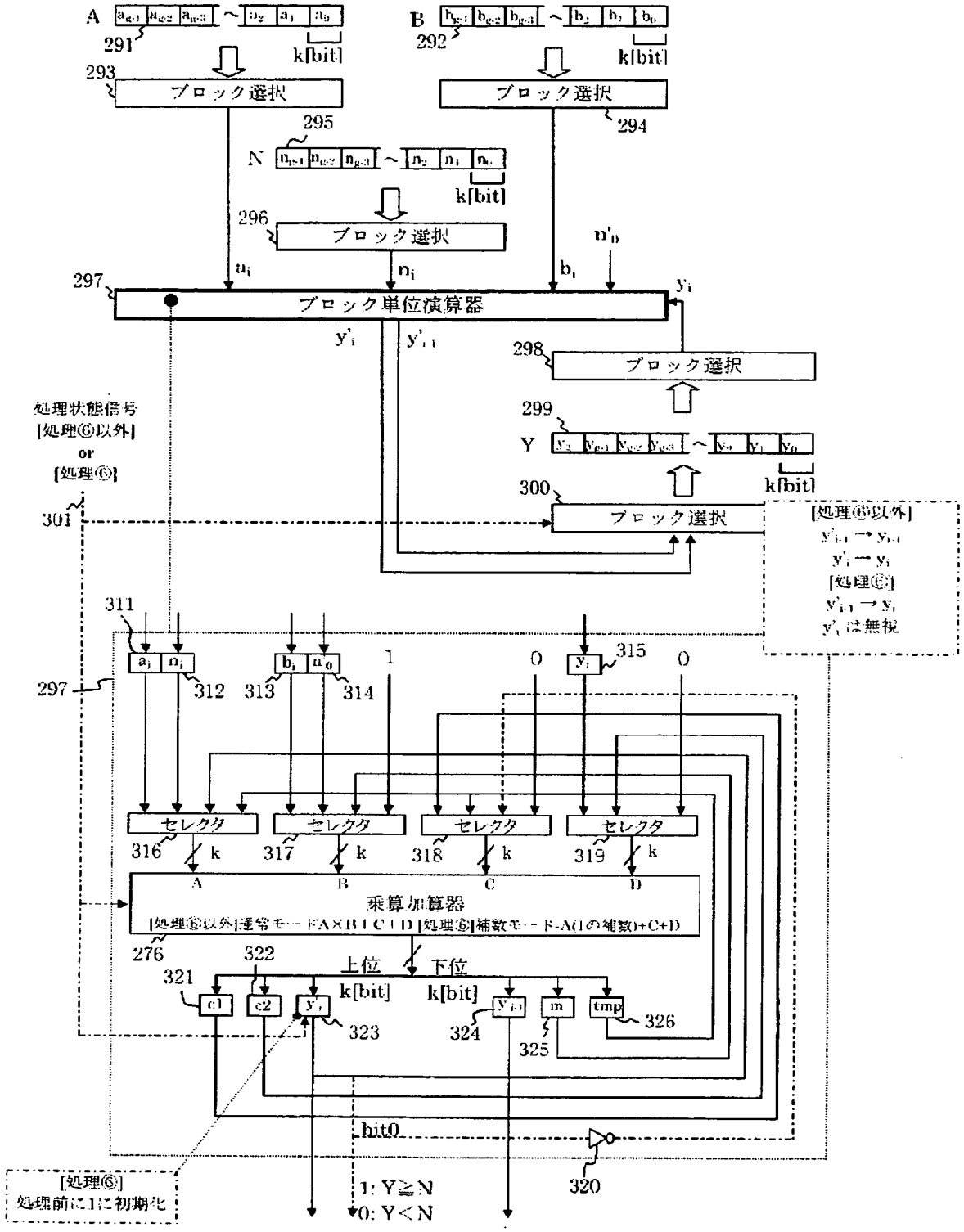
【図９】

第2の多倍長減算回路を示す図



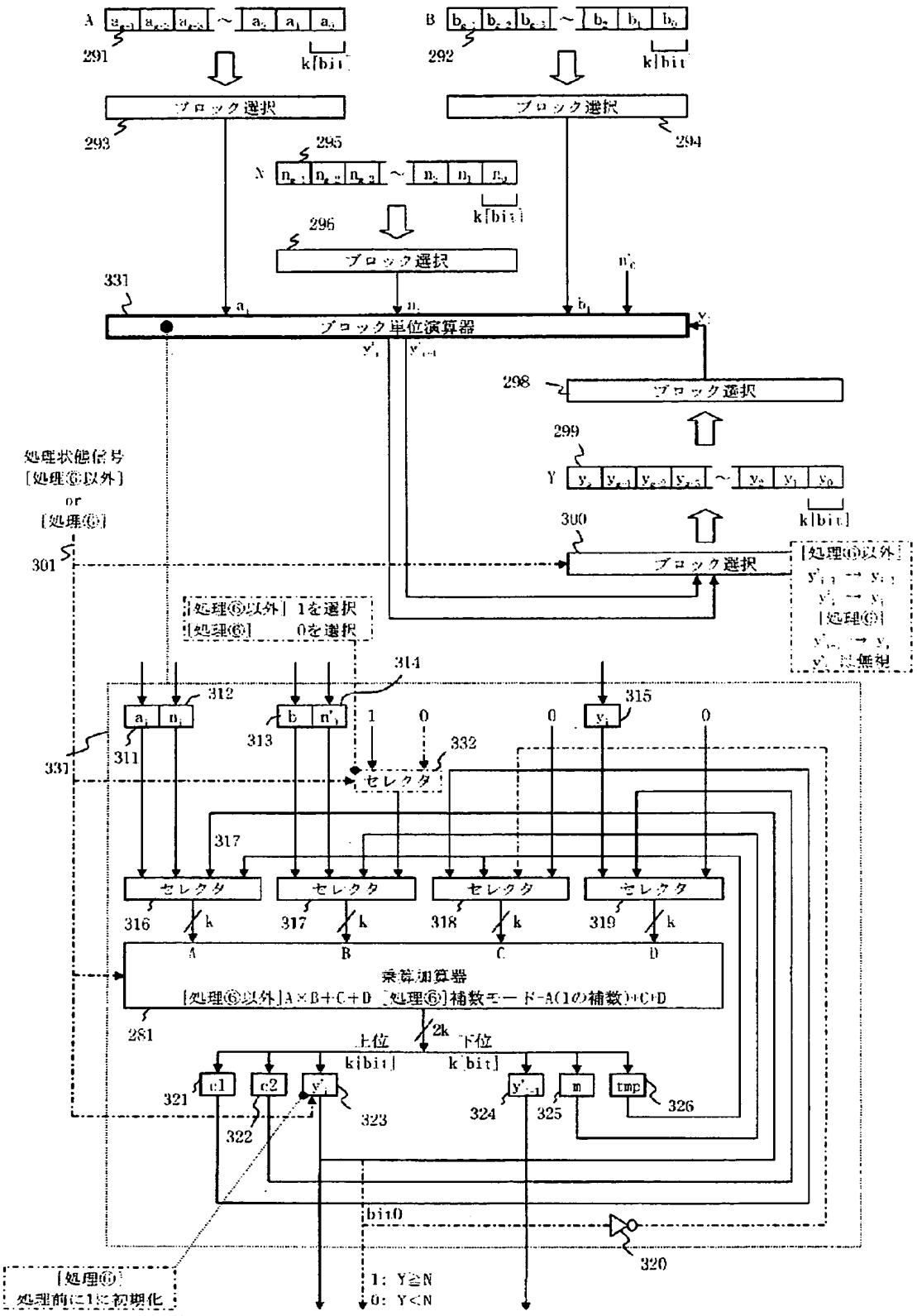
【図１０】

第1の多倍長演算回路を示す図



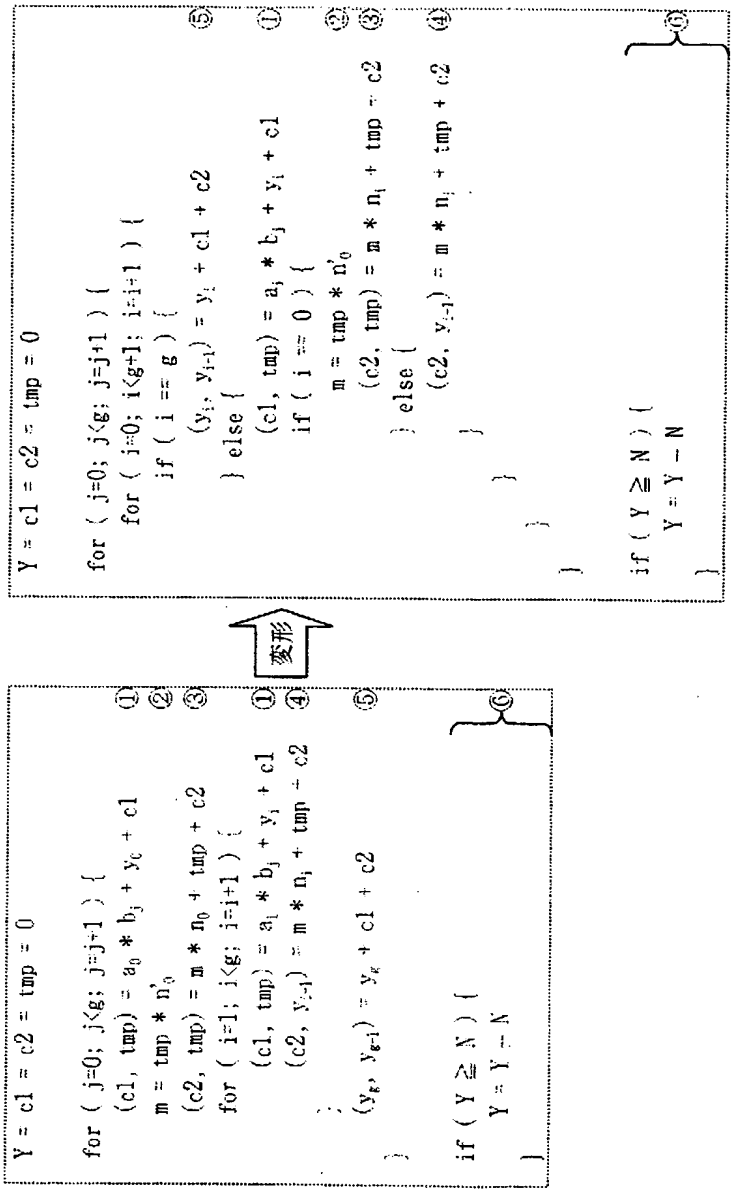
【図１１】

第2の多倍長演算回路を示す図



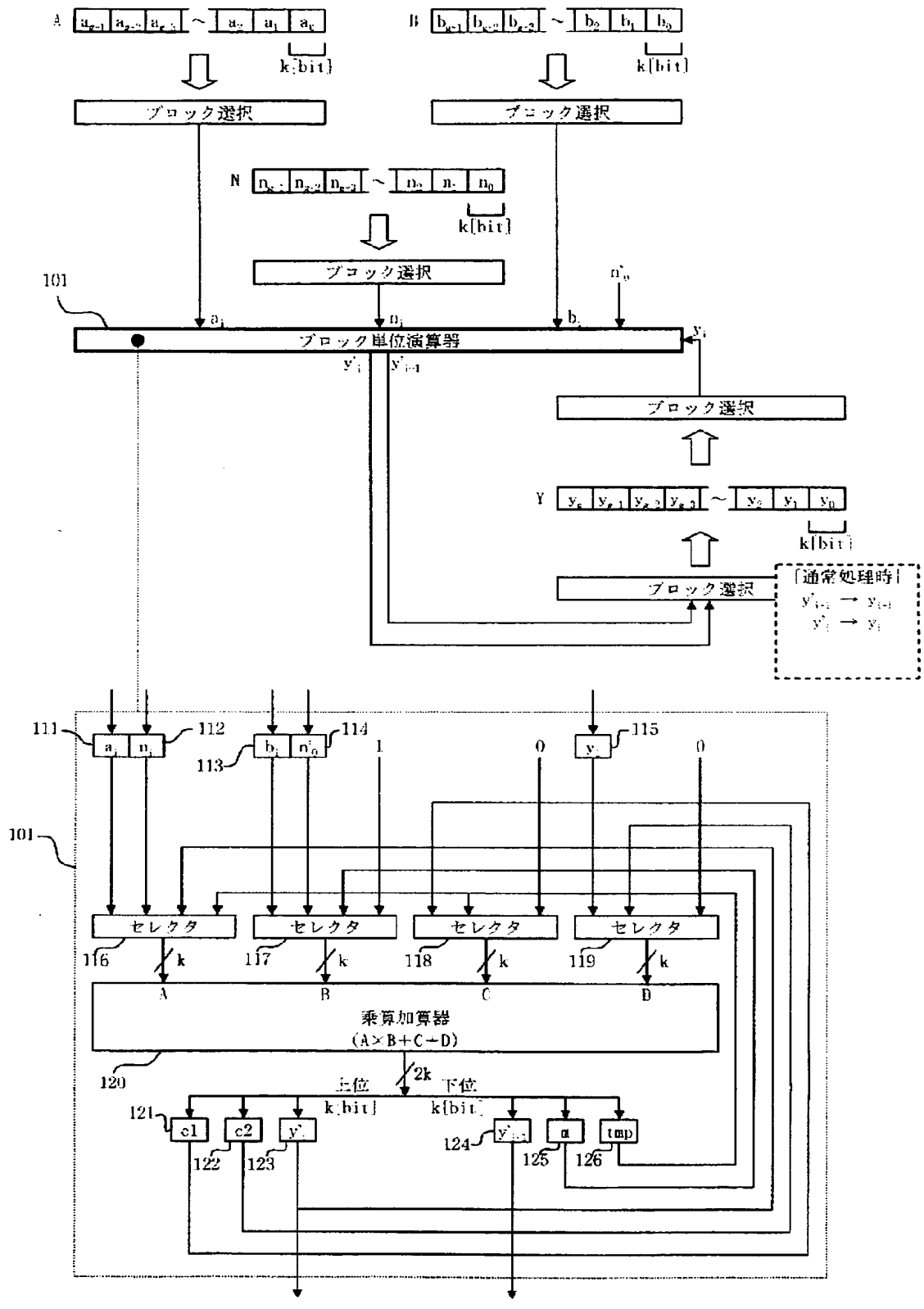
【図１２】

モンゴメリ乗算剰余の多倍長演算アルゴリズムを示す図



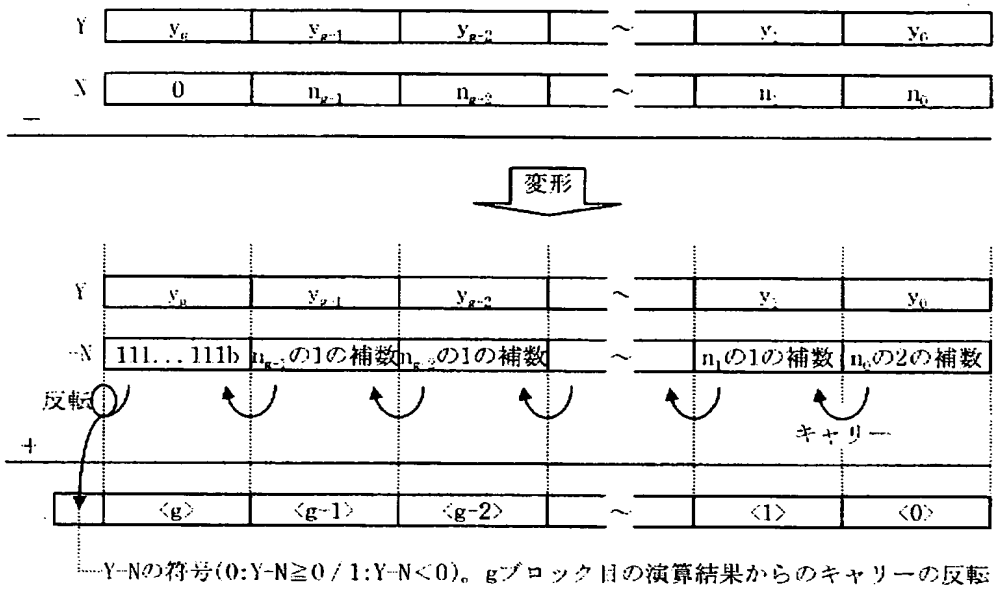
【図１３】

仮想的な乗算剰余演算回路を示す図



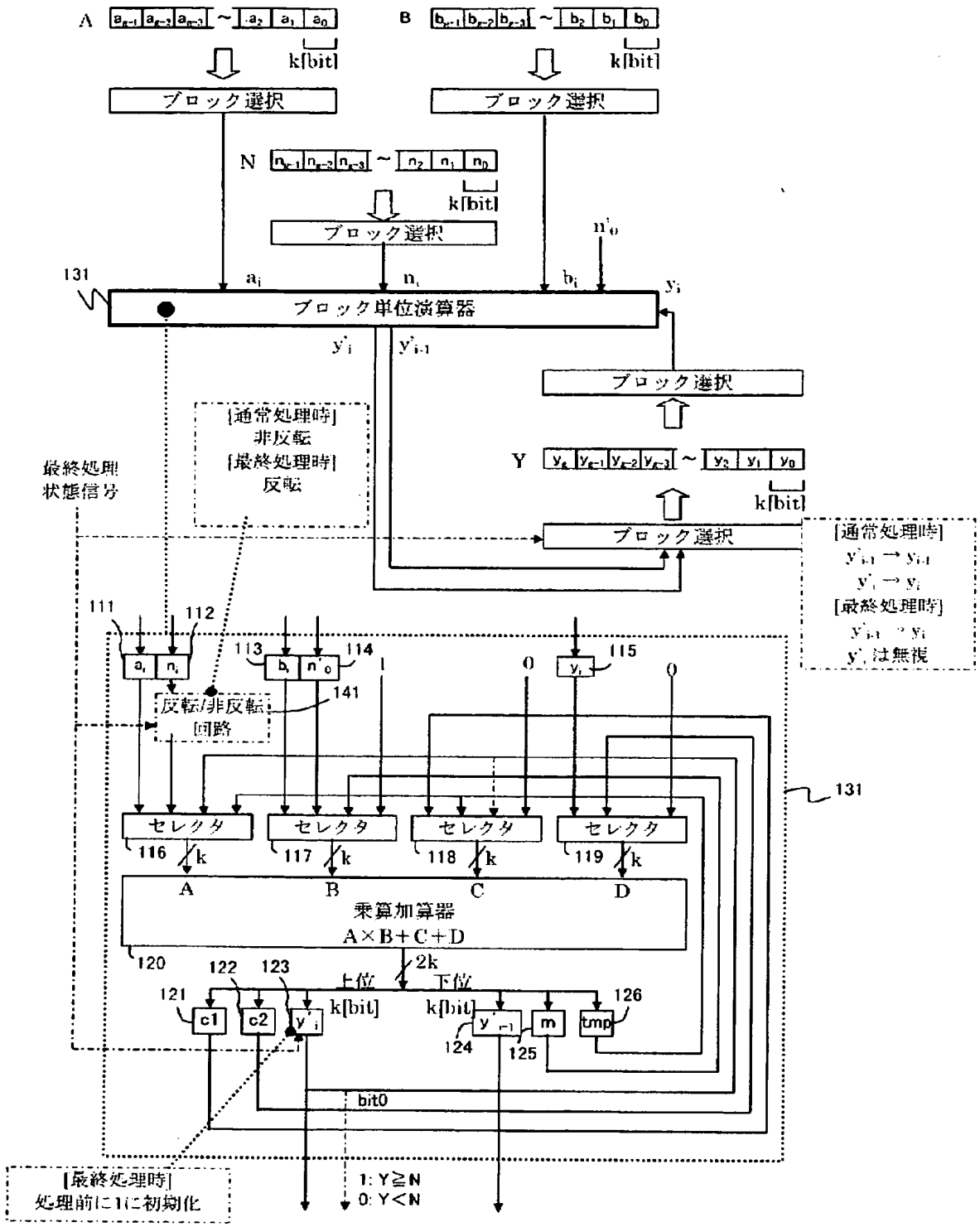
【図１４】

多倍長減算を示す図



【図１５】

仮想的な多倍長演算回路を示す図



【書類名】　　要約書

【要約】

【課題】　モンゴメリ乗算剰余の多倍長演算を行う回路において、ブロック単位演算器における減算のための遅延時間を短縮し、動作周波数を維持したままで演算を行う。

【解決手段】　補数モードにおいて、２次Ｂｏｏｔｈアルゴリズムのエンコーダ手段２０２は、乗算Ａ×Ｂを行う通常モードとは異なる選択信号を出力し、選択手段２０３は、Ｂの３つのビットb_1、b_0、およびb_{-1}に対しては－Ａを表す部分積を選択し、それ以外のビットに対しては０を表す部分積を選択する。加算手段２０４は、－Ａを表す部分積と０とを加算して、－Ａ（Ａの２の補数またはＡの１の補数）を出力する。

【選択図】　　図１

特願２００３－０４６５２７

出　願　人　履　歴　情　報

識別番号　　［０００００５２２３］

１．変更年月日　　１９９６年　３月２６日
［変更理由］　　住所変更
住　所　　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　　富士通株式会社